IMPROVEMENTS IN OR RELATING TO PACKET SWITCHES

The present invention relates to improvements in or relating to packet switches, and is more particularly concerned with cross-bar switches having a cell-level scheduling scheme for handling multicast traffic therein.

Data is transferred over the Internet by means of a plurality of packet switches in accordance with a standard protocol known as Internet Protocol (IP). IP is a protocol based on the transfer of data in variable sized portions known as packets. All Internet traffic involves the transportation of packets of data. Packet switches are devices for accepting incoming packets; temporarily storing each packet; and then forwarding the packets to another part of the network. In particular, a packet switch receives packets of data on a plurality of input ports and transfers each packet to a specific one of a plurality of output ports. The packets of data can be of variable length or of fixed length.

Traffic volume in the Internet is growing exponentially, almost doubling every 3 months. The current capacity of internet protocol (IP) routers or packet switches is insufficient to meet this demand and hence there is a need for IP routers that can route IP traffic at extremely large aggregate bandwidths in the order of several Terabit/s. Such routers are termed "Terabit Routers".

Two important trends are also evident. First, operators are consolidating all traffic onto a single IP back-bone. Secondly, IP is increasingly required to support real-time and multimedia traffic. This means that the next generation of routers must also support 'Quality of Service' (QoS). In particular, they must support low bounded delay for real-time traffic.

The packets transferred in accordance with IP can (and do) vary in size. Within routers it has been found useful to pass data in fixed sized units. In routers the data packets are partitioned into small fixed sized units, known as cells.

One suitable technique for implementing a scalable communications path is a backplane device, known as a cell based cross-bar. Data packets are partitioned into cells by a plurality of ingress means for passage across the cross-bar.

The plurality of ingress means provide respective interfaces between incoming communications channels carrying incoming data and the backplane. Similarly, a plurality of egress means provide respective interfaces between the backplane and outgoing communications channels carrying outgoing data.

A general terabit router architecture bears some similarity to conventional router architecture. Packets of data arrive at input port(s) of ingress means and are routed as cells across the cross-bar to a predetermined egress means which reassembles the packets and transmits them across its output port(s). Each ingress means maintains a separate packet queue for each egress means.

The ingress and egress means may be implemented as line interface cards (LICs). Since one of the functions regularly undertaken by the ingress and egress means is forwarding, LICs may also be known as 'forwarders'. Further functions include congestion control and maintenance of external interfaces, input ports and output ports.

In a conventional cell based cross-bar each ingress means is connected to one or more of the egress means. However, each ingress means is only capable of connecting to one egress means at any one time. Likewise,

each egress means is only capable of connecting to one ingress means at a time.

All ingress means transmit in parallel and independently across the cross-bar. Furthermore cell transmission is synchronised with a cell cycle, having a period of, for example, 108.8ns.

The ingress means simultaneously each transmit a new cell with each new cell cycle.

The pattern of transmissions from the ingress means across the cross-bar to the egress means changes at the end of every cell cycle.

The co-ordination of the transmission and reception of cells is performed by a cross-bar controller.

A cross-bar controller is provided for efficient allocation of the bandwidth across the cross-bar. It calculates the rates that each ingress means must transmit to each egress means. To support multicast traffic, it is also necessary to calculate the multicast rates from ingress means to all relevant egress means. This is the same as the rate at which data must be transmitted from each packet queue. The calculation makes use of real-time information, including traffic measurements and indications from the ingress means. The indications from the ingress means include monitoring the current rates, queue lengths and buffer full flags. The details of the calculation are discussed more rigorously in the copending UK Patent Application Number 9907313.2 (docket number F21558/98P4863).

The cross-bar controller performs a further task; it serves to schedule the transfer of data efficiently across the cross-bar whilst maintaining the calculated rates. At the end of each cell cycle, the cross-bar controller communicates with the ingress and egress means as follows. First, the cross-bar controller calculates and transmits to each ingress means the identity of

the next packet queue from which to transmit. Secondly, the cross-bar controller calculates and transmits to each egress means the identity of the ingress from which it must receive.

By allowing many egress means to receive from the same ingress means at the same time, multicast replication can be achieved.

In accordance with one aspect of the present invention, there is provided a method of operating a packet switch which comprises a plurality of ingress means, a plurality of egress means, a cross-bar and a controller, the cross-bar being connected between the ingress means and the egress means to transfer multicast and unicast data traffic from the ingress means to the egress means, the method comprising the steps of:-

a) determining if the data traffic to be transferred is unicast or multicast;

b) if the data traffic is unicast, invoking a unicast schedule;

c) if the traffic is multicast, invoking a multicast schedule; and

d) transferring the data traffic in accordance with the invoked schedule.

Advantageously, step c) comprises forming a multicast cell fanout table containing current fanout requirements for a cell at the head of a multicast queue in each ingress means.

Step c) further comprises setting eligible bits for multicast cells which are currently allowed to be scheduled.

Step c) further comprises determining a priority for each ingress means for sending the cells.

The method further comprises the step of e) filling a blank multicast schedule in accordance with the priority assigned to each ingress means.

Step e) comprises the step of:-

(i) filling the blank schedule with the full fanout of the first priority ingress means.

Step e) further comprises the step of:-

(ii) filling in as much of the fanout of the next priority ingress means and subsequent ingress means as possible to complete the schedule.

Step (ii) comprises selecting fanouts of ingress means in accordance with multicast egress credit allocated to each egress means.

The term 'fanout' as used herein refers to set of egress means to which the current cell must be replicated.

For a better understanding of the present invention, reference will now be made, by way of example only, to the accompanying drawings in which:-

Figure 1 illustrates multicast ingress rate resolution; and

Figure 2 illustrates multicast scheduling.

The desired features for a multicast scheme in accordance with the present invention are listed below:

- High efficiency (aim for 70-80% pure multicast traffic or better)
- Use shared cross-bar for replication
- Easily integrated into the unicast cell level scheduling algorithm
- Fair across ingress ports to each egress port
- Supports real time and non real time multicast services
- Maintains unicast bandwidth commitments

It is assumed that there is one multicast IP packet (or fixed length part of such a packet) available to be sent at each ingress line interface card (LIC). Although the present invention will be described with reference to LICs, it will readily be appreciated that it is not limited to using LICs and that any suitable device which provide the ingress and egress functions can be used.

The fanout of a multicast packet is defined to be the set of egress ports (of the cross connect) to which the packet must be replicated. The fanout of the next multicast cell must be known by the central scheduler in order that it can be scheduled across the cross connect.

It is assumed that the fanout information for the next multicast cell to be sent from each ingress port is known.

Once ingress bandwidths have been allocated for multicast traffic, scheduling opportunities must be allocated for all ingress LICs to ensure fair access and to preserve allocated rates. The basic scheduling concept for multicast slots is shown in Figure 1.

Figure 1 illustrates separate ingress lines (Figure 1a) and a multicast schedule (Figure 1b).

The four ingress lines are labelled 0 to 3 as shown on the left. Each line has a rate associated with multicast traffic, that is, high and low priority queues. This rate is represented as a send opportunity, indicated by the arrows, every fixed number of cell periods. Ingress line 1 has the highest rate in that it provides a send opportunity every two cell periods. Ingress line 0 has a rate which provides a send opportunity every four cell periods. Ingress line 2 has the same rate as ingress line 0 but is out of phase with it by a cell period. Ingress line 3 has the lowest rate – providing a send opportunity every sixteen cell periods.

The send opportunities of ingress lines 0 to 3 as shown in Figure 1a are combined into a multicast schedule as shown Figure 1b. In accordance with the present invention, the send opportunities are combined by placing a send opportunity on the next free cell cycle (cell cycles are numbered 1 to 16 as shown) unless it would overlap with the next send opportunity for the

same ingress LIC. This means that each ingress has send opportunities as shown in Table 1 below:-

Table 1.

	Cell cycles															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0			√				√				√				√	
1	√		√		√		√		√		√		√		√	
2		√				√				√				√		
3								√								

However, in order to combine these send opportunities in accordance with the present invention, this means that each ingress will send in the free cell cycles as shown in Table 2:-

Table 2.

	Cell cycles															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0			√				√					√			√	
1	√			√	√			√		√		√	√			
2		√				√					√			√		
3									√							√

It can readily be seen from Table 2 above and from Figure 1b that cell period 12 has two send opportunities where the send opportunity for ingress LIC 1 has to be stacked on top of the cell send opportunity for ingress LIC 0 to avoid it colliding with its next send opportunity. This means that, for cell 12, LIC 1 must be transmitted before LIC 0, that is, LIC 1 is first choice and LIC 0 is second choice. It will be appreciated that LIC 1 has to be first choice as it will collide with itself on the next available cell cycle.

The net effect of this process is to spread the multicast send opportunities on as many cell periods as possible thus reducing the height of

each stack. The height of each stack is directly related to the number of ingress multicast cells that have to be scheduled in this cycle and thus the amount of sorting that has to be carried out by the algorithm.

Whilst minimising the amount of work to be carried out in each cell cycle by the multicast scheduler, the maximum jitter is also limited to 1/rate. A cell send opportunity is never delayed by more than the gap between the ideal ingress cell send opportunities, thus making the jitter inversely proportional to the rate. Multicast real time delay jitter can thus be improved by allocating higher rates.

If there is no multicast cell send opportunity scheduled, then a normal unicast scheduling algorithm is invoked. If there is one or more multicast cell send opportunities then a multicast scheduling algorithm is invoked in accordance with the present invention.

Figure 2 shows the active components in the multicast scheduling algorithm. A multicast cell fanout table is shown in Figure 2a which contains the current fanout requirements for the cell at the head of the multicast queue in each ingress LIC. The table comprises an ordered list of pointers relating to the stack of multicast send opportunities on a particular cell period. The ingress LICs are listed on the left of the table and the egress LICs are listed on the top of the table. As shown, ingress 0 is sending to egresses 0, 2, 4, 6, 8 and 10, ingress 1 is sending to egresses 2, 3, 4, 9 and 10, ingress 3 is sending to egresses 9 and 10, and ingress 6 is sending to egresses 5, 6, 9 and 10.

When the multicast schedule as shown in Figure 1b is carried out for these specific examples, it is found that ingress 1 is the first choice and ingress 6 is the second choice. This means that ingress 1 has first priority to send a multicast cell followed by ingress 6.

Each egress has a multicast credit as shown in Figure 2b. The multicast egress credit is determined in accordance with a multicast bandwidth allocation method as described in co-pending British patent application no. 0018328.5 (docket number 2000P04909). The egress credit is the maximum number of multicast cells that can be sent to each egress in one bandwidth allocation period (BAP). A BAP is the number of cell cycles over which the calculated rate remains static, that is, the calculated rate may change ever BAP. As shown in Figure 2b, egress 0 has a credit of 27, egress 1 has no credit and egresses 2 and 3 have credits of 35 and 5 respectively.

As the algorithm supports the concept of scheduling part of the fanout and leaving residues, the entries in the multicast cell fanout table will change, that is, 1s will go to 0s when they have been scheduled, when part of the fanout has been scheduled. Another table (not shown) is thus required to remember the full fanout for the duration of the packet. This second table, called the multicast packet fanout table, is updated when the next fanout is sent to a traffic management card in the controller at the end of the packet.

The algorithm starts with a blank schedule and fills this with the full fanout of the first choice ingress LIC, as shown on the right of Figure 2d in the compilation of the control frame. In this case, ingress LIC 1 is the first choice and 1s have been entered in the control frame corresponding to egresses 2, 3, 4, 9 and 10 leaving the remaining positions blank. It then moves to the next ingress LIC of choice and schedules as much of the fanout as possible. In the example illustrated in Figure 2a, the second choice is ingress LIC 6. In this case, as egress LICs 9 and 10 have been taken up with the fanout for the first choice, only egress LICs 5 and 6 can be added to the control frame. This is shown as 6s in the control frame of Figure 2e in the positions corresponding to egress LICs 5 and 6.

It will be appreciated that this will be repeated for subsequent choice ingress LICs until the control frame is as full as it can be. For example, if ingress LIC 0 is the third choice, 0s will be added to the control frame to correspond to egress LICs 1 and 8. Ingress LIC 3 cannot add anything to the control frame as the two egress LICs to which it is to multicast are already taken by the first choice ingress LIC, that is, ingress LIC 1. Although ingress LIC 2 has a cell which can be transmitted to egress LIC 1, this cannot be added to the control frame as there is no multicast egress credit as will be described more fully below. A fully compiled control frame for the example shown in Figure 2 may comprise the following:-

| 0 | | 1 | 1 | 1 | 6 | 6 | | 0 | 1 | 1 | |

For the multicast cells that are scheduled for this cell period, that is, cells from ingress LICs 1 and 6 in this example, the eligible bits are set. As shown in the 'eligible' status column (Figure 2c), ingresses 0, 1, 2 and 6 are eligible for multicast as shown by the 1s. Ingress 3 is not eligible for multicast as shown by the 0. These bits are only reset when the cell has been scheduled for the complete fanout. In the example, shown in Figure 2, only ingress LIC 1 will be reset as that is the only ingress LIC which has been completely scheduled for the whole fanout.

The algorithm will then attempt to schedule as many of the other multicast cells as possible, starting with the one whose ingress LIC number is one larger than the first choice, subject to two constraints. The first is that the eligible bit is set, the second is that the multicast egress credit is positive for a particular egress destination. The attempt to schedule as many other multicast cells as possible is cyclic in nature and once all ingress lines in the multicast cell fanout table have been evaluated for scheduling from the first

choice ingress line, the algorithm returns to the top of the fanout table and carries on until all ingress line entries have been evaluated.

Any gaps in the compiled frame can be filled by unicast traffic.

The multicast egress credit is decremented each time a multicast cell is scheduled for that particular egress. In a real implementation there will probably be separate credit for real time multicast cells and for non real time multicast cells. This credit can be refreshed by the equivalent allocated egress rates every BAP period or on a sub-multiple of a BAP period depending on the amount of egress smoothing desired.